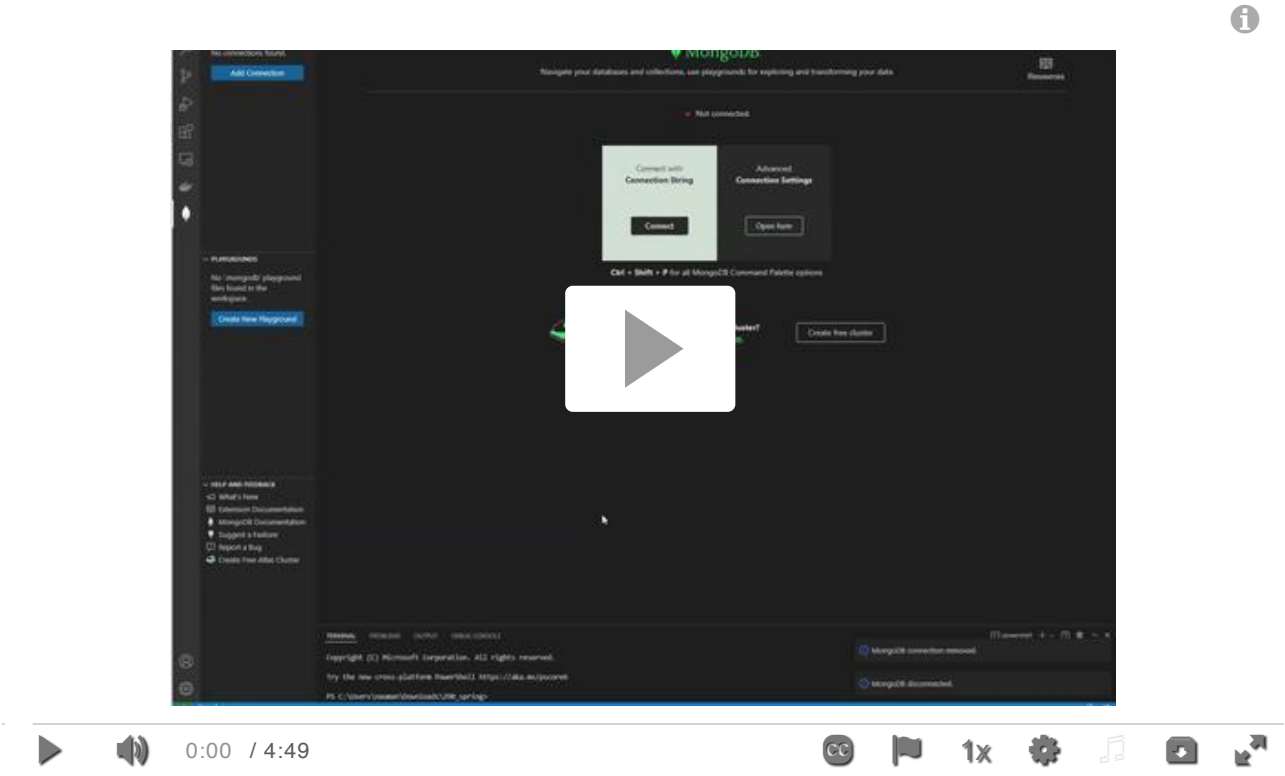
Exploration — Interacting with MongoDB

Introduction

Our main goal is to write Node applications that can interact with MongoDB. However, in this exploration, we will look at different non-programmatic interfaces that we can use to interact with MongoDB. We will look at 3 different interfaces: the VS Code MongoDB extension (recommended), mongosh which is a text-based command line interface, and a GUI (Graphical User Interface) provided by MongoDB Cloud.

VS Code MongoDB Extension

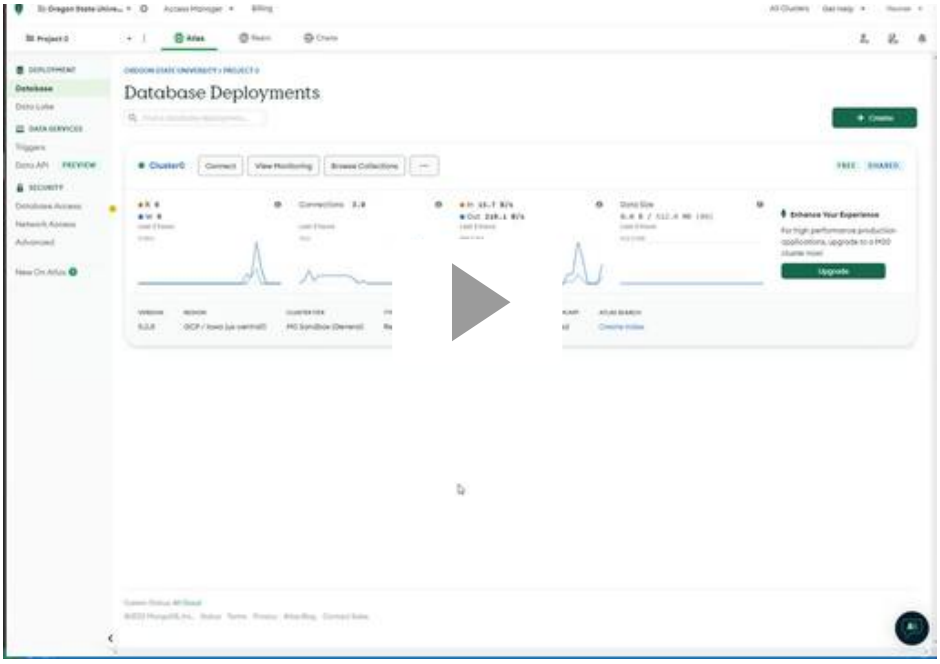
Option 1: A MongoDB extension is available for VS Code. We can install this extension and use it to interact with a MongoDB server. This extension though supports a subset of the functionality provided by mongosh and the MongoDB Cloud GUI. However, when we are using VS Code to write and debug an application that perform CRUD operations on a MongoDB server, this extension allows us to easily verify the results of our app's CRUD operations.



mongosh: The MongoDB CLI

Option 2: MongoDB has a command line interface (CLI) called **mongosh**. We can download and install mongosh following directions at the **Install mongosh (https://www.mongodb.com/docs/mongodb-shell/install/)** page.

The commands used in the following video are available in **this text file (https://canvas.oregonstate.edu/courses/1879154/files/93832000?wrap=1)** (https://canvas.oregonstate.edu/courses/1879154/files/93832000/download?download_frd=1) . If you have problems installing mongosh on your machine, you can instead use the MongoDB Cloud GUI to interact with your MongoDB Cloud server. Use of that GUI is described in the next section of this exploration.



0:00 / 15:15

Here are some useful commands we can run using mongosh:

Command	Description
`show dbs`	List all databases in the MongoDB server
`use db_name`	Use the database `db_name`. Substitute `db_name` with the name of database you want to connect to. If the database does not exist, it will be created.
`show collections`	List all the collections in the current database
`exit`	Exit mongosh

JavaScript API

mongosh includes a JavaScript API which we can use to create and manage collections, as well as perform CRUD operations on a collection. A **cheat-sheet at MongoDB website (https://developer.mongodb.com/quickstart/cheat-sheet/)** provides a helpful guide to these functions.

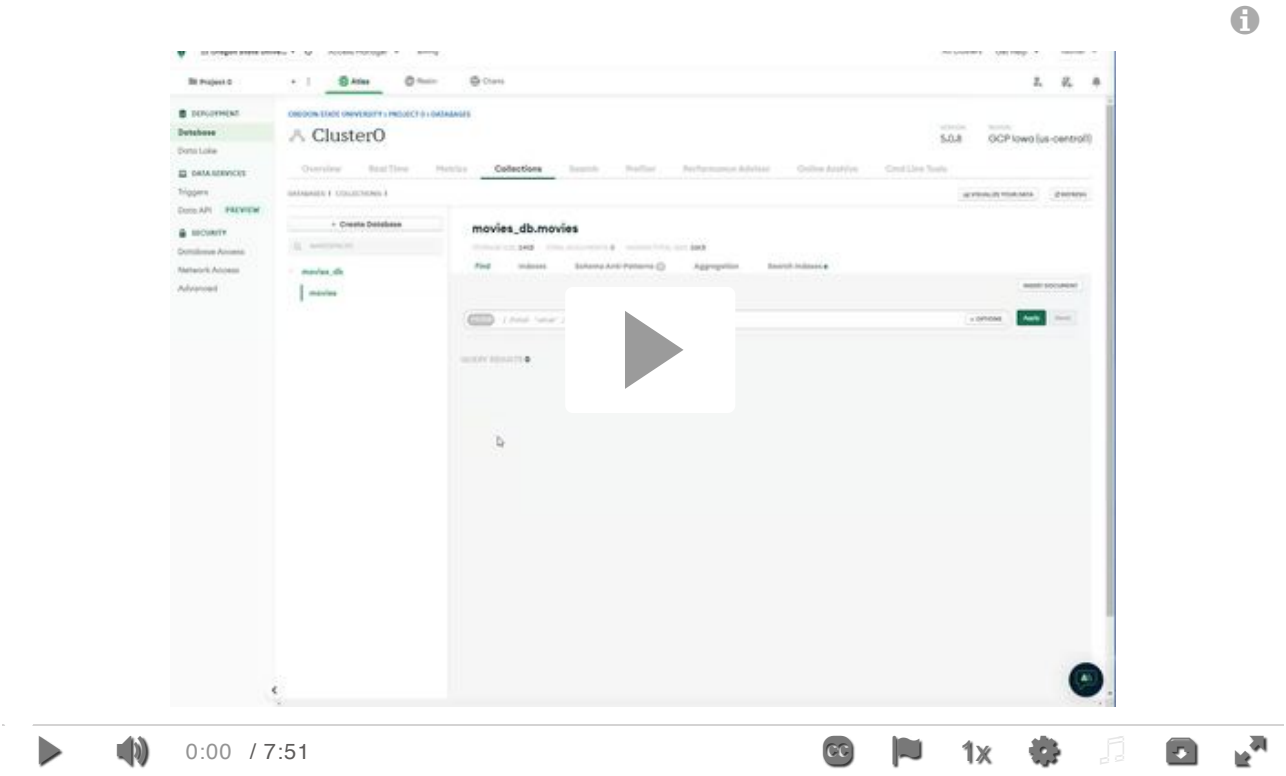
Command	Description
`db.getCollectionNames()`	List all the collections in the current database
`db.createCollection(...)`	Create a new collection

The available CRUD operations on a collection are specified **here in the cheat-sheet (https://developer.mongodb.com/quickstart/cheat-sheet/#crud)**. Note that you need to replace `coll` in the name of these functions with the name of your collection.

Command	Description
`db.coll.insertOne(...)`	Create a document in the collection `coll`. The argument is a JavaScript object. MongoDB creates and associate a unique object ID value with each document. This value is stored in the property `_id`.
`db.coll.findOne()`	Return a single document at random from `coll`.
`db.coll.findOne(...)`	Return a single document from `coll` based on the criteria passed in the argument.
`db.coll.find()`	Return all documents from `coll` that match the criteria passed in the argument. If the number of documents exceeds 20, then batches of up to 20 documents are returned. Type `it` to get the next batch
`db.coll.updateOne({"_id": ObjectId("123")}, {...} )`	Update the document with object ID `123`.
`db.coll.deleteOne({"_id": ObjectId("123")})`	Delete the document with object ID `123`.
`db.coll.deleteMany({ ... })`	Delete all the documents matching the criteria passed in the argument.

The MongoDB Cloud GUI

Option 3: The MongoDB Cloud GUI provides us with a no-install UI to interact with our MongoDB Atlas server running on MongoDB cloud.



Summary

In this exploration, we discussed three different tools that allow us to interact with a MongoDB server without writing programs. We first looked at mongosh which is command line interface (CLI) for running commands on a MongoDB server. mongosh includes a JavaScript API for all CRUD operations. We then looked at a GUI provided at the MongoDB cloud website. Finally, we looked at the VS Code MongoDB extension.

Additional Resources

Here are some references to learn more about the topics we discussed in this exploration.

- **MongoDB website** (https://docs.mongodb.com/) has extensive documentation including **a cheat-sheet for mongo commands** (https://developer.mongodb.com/quickstart/cheat-sheet/) .
- For more information on VS Code MongoDB Extension see **Working with MongoDB** (https://code.visualstudio.com/docs/azure/mongodb) page on VS Code website and **MongoDB For VS Code** (https://www.mongodb.com/products/vs-code) page on MongoDB website.