CSC561 NoSQL Databases
Lab 3, Part 3: Document Database / MongoDB Part 3

Part 3: Access from Programming Language (Node.js and Mongoose)

The tasks we want to complete in this section are to:

- Access our MongoDB database from a Node.js Mongoose app. Node.js is a server side JavaScript run-time environment
- Perform all CRUD methods for api and web calls from Node.js using Mongoose which is an Object-Relational Mapping (ORM) for MongoDB

Utilizing a database might be all that you need in the real world, but accessing that database from a programming language such as JavaScript will give you an advantage in the workplace. Knowing how to retrieve, manipulate and save data from a programming language is not a new feature of any programming language or database, so it will be helpful to understand how to set-up and perform some basic tasks.

In this lab, we will:

1. Map a drive to the Windows share on the Docker container with the Node.js code
2. Open up the Node.js project in the Visual Studio code IDE
3. Implement the user.js model by creating the methods (getUserById, addUser, updateUser, removeUser) to execute the MongoDB queries (findById, create, findOneAndUpdate, deleteOne) respectively using the Mongoose ORM. These methods will then make the web and api routes work on your application. The rest of the code is already provided to you.

Video Tutorials

Please review these video lectures:

Learning Node.js
https://www.lynda.com/Node-js-tutorials/Welcome/612195/677534-4.html?org=uis.edu (You may need to click the Sign In button in the upper right corner, and then log in with your UIS NetID and password.)

Access from Programming Language (Node.js Mongoose ORM)
[68 min 05 sec]:
https://cdnapisec.kaltura.com/index.php/extwidget/preview/partner_id/1371761/uiconf_id/13362791/entry_id/1_b88z1mj2/embed/dynamic

Work on your own to submit

This lab relies on the work done in Lab3 Part 1, so please be sure you have completed those steps before moving on to Part 3.

Use this table to determine which container is yours. You will log into the share with .\NetID for the username (.\tllos1 for example) and your UIN for the password.

Netid	Windows share	Url of the PHP application
agang2	\\10.64.3.56\agang2	https://csc570e.uis.edu:9444
bbala5	\\10.64.3.56\bbala5	https://csc570e.uis.edu:9445
bguti6	\\10.64.3.56\bguti6	https://csc570e.uis.edu:9446
brodr22	\\10.64.3.56\brodr22	https://csc570e.uis.edu:9447
chick7	\\10.64.3.56\chick7	https://csc570e.uis.edu:9448
eunsik2	\\10.64.3.56\eunsik2	https://csc570e.uis.edu:9449
jlund6	\\10.64.3.56\jlund6	https://csc570e.uis.edu:9450
jshei3	\\10.64.3.56\jshei3	https://csc570e.uis.edu:9451
mpavl3	\\10.64.3.56\mpavl3	https://csc570e.uis.edu:9452
rsayy2	\\10.64.3.56\rsayy2	https://csc570e.uis.edu:9453
sarya7	\\10.64.3.56\sarya7	https://csc570e.uis.edu:9454
skoch7	\\10.64.3.56\skoch7	https://csc570e.uis.edu:9455
szhen6	\\10.64.3.56\szhen6	https://csc570e.uis.edu:9456
zwold2	\\10.64.3.56\zwold2	https://csc570e.uis.edu:9457
smeka6	\\10.64.3.56\smeka6	https://csc570e.uis.edu:9458
pshet7	\\ 10.64.3.56\pshet7	https://csc570e.uis.edu:9459

You will need to update the server.js file with the IP address of your MongoDB VM. Ex:

```
mongoose.connect('mongodb://10.92.128.109:27017/blogger', { useNewUrlParser: true });
```

In order to implement the user.js model you will need to look up the Mongoose documentation (https://mongoosejs.com/docs/api.html) and figure out what parameters each method takes and what it returns.