# Module 3 Project 2

Michael Grandori

Arizona State University

IFT: 458

Professor Dinesh Sthapit

September 22, 2022

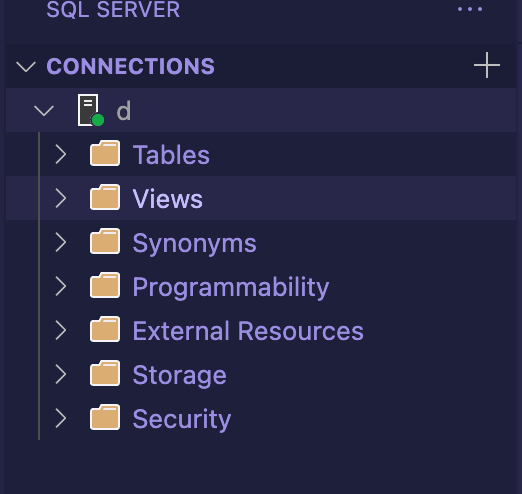
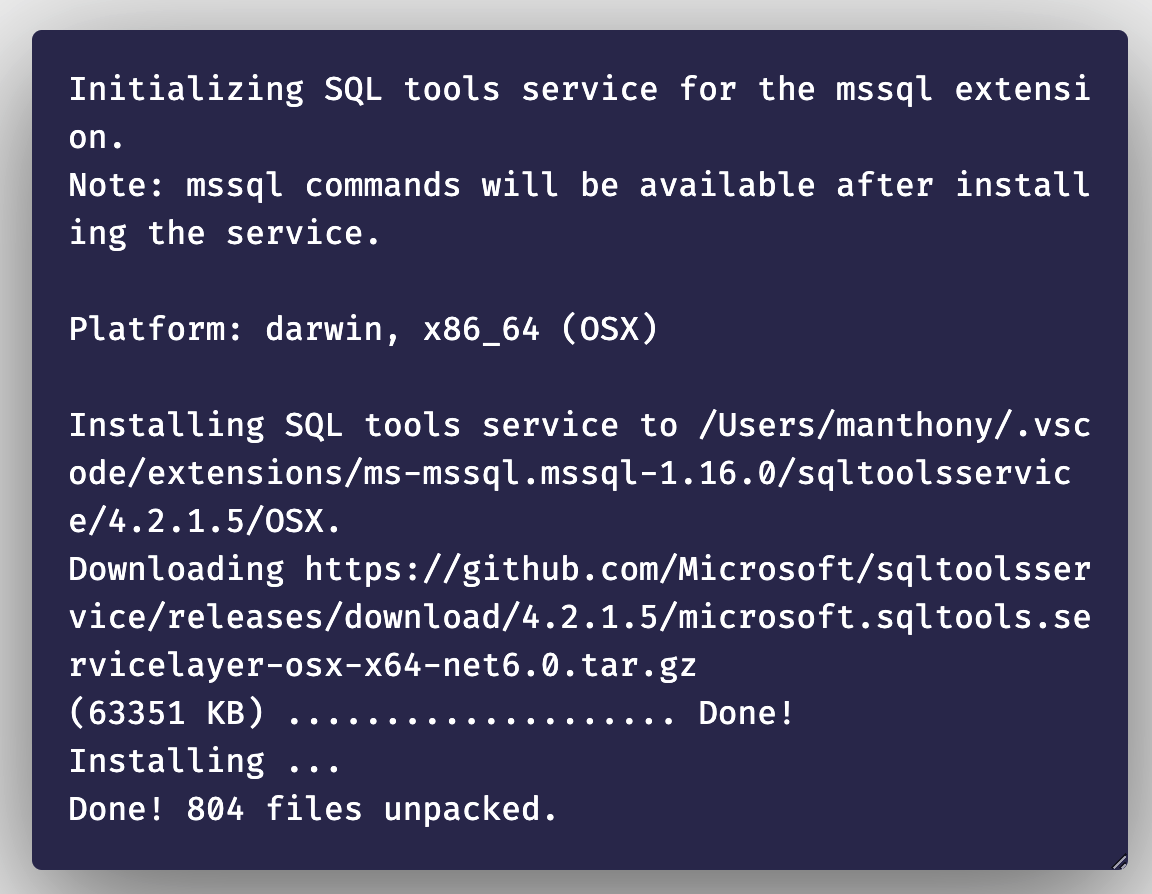
In this assignment, we focused on creating a connection to a database from our server, and then subsequently using a router, a controller, and a db manager to be able to create an API to get customer data from our DB. We used the http tool Postman to check our route to get all customers from our table.

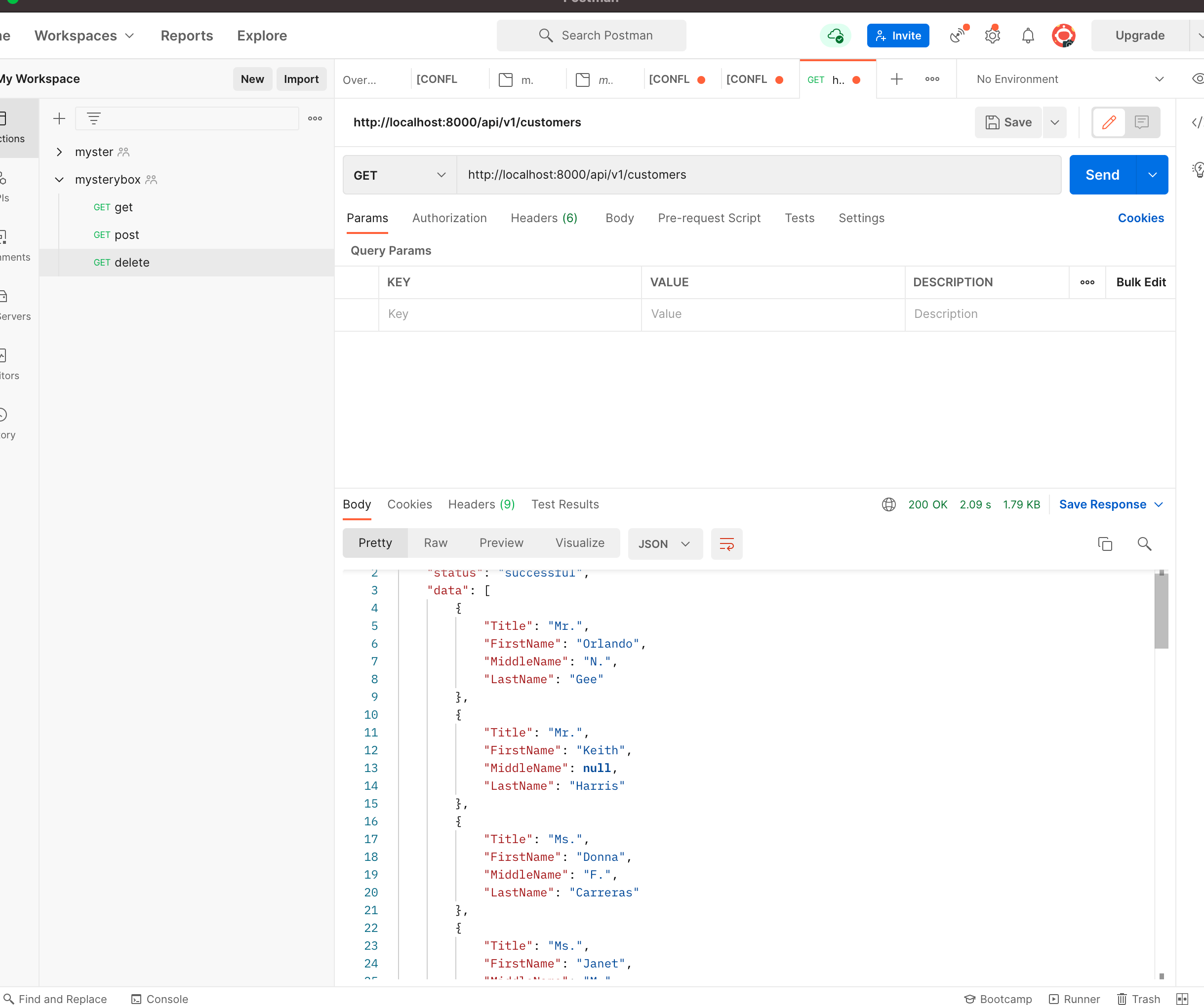
To be able to connect to a sql server, you need to have the correct credentials. In our case, we used an extension in VS Code as a client to be able to make this connection to a hosted SQL server instance. After installing the extension, you can use the credentials such as server, database name, user and password to establish a connection. When the connection is established, you can view the tables and relationships that exist on the DB you are connected to.

We then began our Node Js server app. For this, we needed to install some important packages and middleware for express which acts as our server middleware and makes handling routing and connections easier. We created a routers directory and within it pass our routes that we need onto the router object that express exposes. This is a singleton object that you can keep appending routes onto so that if the incoming request route matches, it will pass it to the controller to handle. The controller’s job is to call the db managers function and then return a status code along with a response, whether it be the requested data or an error with the request. The DB manager’s job is to create a connection to the database, fetch the data using SQL, and then return it to the controller for it to formulate a response. By splitting up these things it makes it very modular and not dependent on one another. It also allows you to easily swap out databases or any layer within since they do not depend on each other as if it was one large function that matches the route, does logic, connects to db, and returns a response. This would be a much harder system to maintain.

We finally used Postman http client to test our route. When hitting ‘http://localhost:3000/api/v1/customers` with a GET, my query would return the main fields (I did not want to expose all the customer data to the outside world in a request!) top 20 results to me in JSON format.

**Screenshots**

****

****

