

CS591 MEAN Homework 2

In this assignment you will extend the functionality of Homework 1 to include a mongo database, using the mongoose ODM package. The app will still return the length of a string passed using either a GET or POST, and adds a feature to return all stored strings and their lengths. A DELETE route will delete an entry from the database.

Use Mongoose to create a simple schema to store a string and its length.

1. In your HW1 project, copy the file containing your routes to a new file, hw2.js
2. Mount hw2.js on a new path in app.js, for example /hw2
3. Add lookup and storage functionality to your GET and POST route:
 1. GET: When passing a string on the query (i.e. <http://localhost:3000/hw2/longstring>), first look in the database to see if the string is already present. If it is, return the string and its length as read from the database. If it isn't, compute the length, store the string and its length in the database, and return both to the client.
 2. GET (new route): If no parameter is passed on the URI (i.e. <http://localhost:3000/hw2>), return all strings currently stored in the database.
 3. POST: Similar to the GET, when passed a string, first look in the database to see if the string is already present. If it is, return the string and its length as read from the database. If it isn't, compute the length, store the string and its length in the database, and return both to the client. If no string is passed, return a message in JSON format prompting the user to provide a string.
 4. DELETE: This route takes a string, and if the string is present in the database, it deletes the document and returns a message in JSON format indicating success. If the string is not present, return a 'string not found' message in JSON format.

Use Postman to test you application. There's a sample route file on GitHub in the project/mongoose folder that you might find helpful.

Deliverable: Link to your GitHub repo.